Subject: Re: subset an image programatically
Posted by lbusett on Mon, 02 Jan 2006 10:44:45 GMT
View Forum Message <> Reply to Message

Hi Javier,

the problem in your code seems to be that the ENVI_GET_ROI_DATA command returns only the values of your image for the pixels that are part of the ROI, but not their "position" in the image. If your ROI is formed by 14000 pixels, the "roi_data" variable will be an array with 14000 elements, while the "output" that you want to write to disk is a matrix with the same dimensions of the input image that contains data only for the pixels that are contained in your ROI.

In order to perform your task, you have first of all to create a "black" image with the same dimensions of the original image, then you have to retrieve the position of each of the pixels of your ROI with the "ENVI_GET_ROI" command, and finally you have to put the values that you have retrieved with the ENVI_GET_ROI_DATA command in the correct positions of the "black" image and write it to disk with the ENVI_WRITE_ENVI_FILE command.

I think that something like this should work, at least if the images that you want to subset are "1 band" images.

```
roi_data = ENVI_GET_ROI_DATA( roi_id, fid=fid, pos = [j])
ENVI_GET_ROI_INFORMATION, roi_id, ns=ns, nl=nl
roi_addresses = ENVI_GET_ROI(roi_id)           ; Get the "position" of
the pixels in the ROI

out_data = intarr ( ns,nl,nb)                                   ;
Create the "black" image"

out_data [*,*,*] = -9999     ; Assign a value to all the pixels of the
out image so that their    values will not be "similar" to the values
that you extract from the ROI

out_data[roi_addresses] = roi_data                       ; Assign
values to the pixels of the output image that are in the ROI

ENVI_WRITE_ENVI_FILE, roi_data, ns=ns, nl=nl,      $  ; Write the image
out_name='out_name', map_info=map_info
```

I didn't test it, so maybe it will not work properly.
If you have a single vector file that you are using to subset a series of images with the same dimensions, i also think that you should take a look at this thread:

http://groups.google.com/group/comp.lang.idl-pvwave/browse_t hread/thread/4b90c8f4a2c7b715/

Hope this helps,

Lorenzo

Lorenzo Busetto
Environmental Dynamics Remote Sensing Lab.
University of Milano-Bicocca
email: lorenzo.busetto@unimib.it
tel: 00390264482848

Javier Martinez wrote:

```
>  Hi everyone,
>
>  this is my first post, I'm not a programmer and I'm trying to subset a
>  series of images with vector files in evf format. Looking the envi and
>  IDL help files and some code that I found in this group, I write the
>  code below, and it works properly but not completely. In the final part
>  where I write the envi file I obtain an image with all the roi data in
>  a single line (more than 14000 samples) and I need the data in the
>  original geographical shape (obviously).
>
>  Thanks in advance for any help that you can give me
>
>  happy new year for all you guys
>
>  Javier Martinez Pincheira
>  Instituto del Medio Ambiente
>  Universidad de La Frontera
>
>  pro batch_roi
>  envi,/restore_base_save_files
>  envi_batch_init, log_file='log.txt'
>
>  envi_open_file, 'image', r_fid=fid
>  envi_file_query, fid, ns=ns, nl=nl, nb=nb
>
>  map_info = envi_get_map_info(fid=fid)
>
>  evf_fname = 'vector.evf'
>  evf_id = envi_evf_open(evf_fname)
>
>  envi_evf_info, evf_id, num_recs=num_recs, $
>  data_type=data_type, projection=projection, $
>  layer_name=layer_name
>
>
```

```
>
> FOR i=0,num_recs-1 DO BEGIN
>  ; read the record
>  ;
>  vec = ENVI_EVF_READ_RECORD(evf_id, i)
>  xmap= vec(0,*)
>  ymap= vec(1,*)
>
>  ;just to verify
>  print, 'Number of Records ' + ': ', num_recs
>  print, 'Number of nodes in Record ' + $
>      strtrim(i+1,2) + ': ', n_elements(vec[0,*])
>
>  envi_convert_file_coordinates, fid1, xf, yf, xmap, ymap
>
>
>  roi_id = ENVI_CREATE_ROI(ns=ns, nl=nl, color=4, name='shape')
>
>
>  ENVI_DEFINE_ROI, roi_id, xpts=REFORM(xf, /over), $
>   ypts=REFORM(yf, /over), /polygon
>
>     FOR j=0,0 DO BEGIN
>      ;print, roi_id
>      roi_data = ENVI_GET_ROI_DATA( roi_id, fid=fid, pos = [j])
>     ENVI_GET_ROI_INFORMATION, roi_id, ns=ns, nl=nl
>
>     ENVI_WRITE_ENVI_FILE, roi_data, ns=ns, nl=nl,
> out_name='out_name', map_info=map_info
>
>     ENDFOR
>  ENDFOR
>  print, roi_id
> ENVI_BATCH_EXIT
>  end
```

Subject: Re: subset an image programatically
Posted by Jeff N. on Mon, 02 Jan 2006 20:05:26 GMT
View Forum Message <> Reply to Message

I would drop the call to ENVI_GET_ROI_INFORMATION. The values of ns & nl are set early in the code by a call to ENVI_FILE_QUERY, which is querying the input image. So the call to ENVI_GET_ROI_INFORMATION is both redundant and potentially a source of error if something goes wrong in setting up the ROI's.

Also, I would set up the "black image" like this:

```
out_data = intarr ( ns,nl,nb) - 9999
```

which allows you to skip a construct like this: out_data[*,*,*]

I'm also not sure why you need to bother creating ROI's in the first place? If all that you're doing is setting every pixel value outside of a vector record to -9999 (or whatever value you prefer) and keeping every value inside the vector as is, I think you have everything you need to do this without the ROIs. The variables xf and yf (later REFORMed to xpts and ypts) contain the pixel addresses of the original input image that are inside the vector record, unless I am really missing something. Why not just build a mask for the original image using those?

Jeff

Subject: Re: subset an image programatically
Posted by Javier Martinez on Tue, 03 Jan 2006 19:20:18 GMT

View Forum Message <> Reply to Message

Hi,
thanks for the hints. I'm already try to do the job using the ENVI_MASK_APPLY_DOIT routine, but with this I obtain an image of the same dimensions than the original image but (obviously) with the pixel outside of the vector file masked out, and the thing that I want to do Its a subset image of smaller size than the original, that match the dimensions of the vector file. If you know a way to do this job please let me know.

Thanks again

Javier Martinez

Subject: Re: subset an image programatically
Posted by Jeff N. on Tue, 03 Jan 2006 23:09:29 GMT

View Forum Message <> Reply to Message

Aha. I would do something like this. Find the maximum and minum values of the xpts and ypts arrays, naming them something like min_x, max_x, min_y, max_y. If you'd like, you can add a few extra pixels so that the vector doesn't touch the boundary of the image   here too (for example, min_x = min(xpts) -2) Then do some array subscripting on the input image:

```
out_image = input_image[min_x:max_x, min_y:max_y]
```

Then write out_image out to a file.

Jeff

Javier Martinez wrote:

> Hi,
> thanks for the hints. I'm already try to do the job using the
> ENVI_MASK_APPLY_DOIT routine, but with this I obtain an image of the
> same dimensions than the original image but (obviously) with the pixel
> outside of the vector file masked out, and the thing that I want to do
> Its a subset image of smaller size than the original, that match the
> dimensions of the vector file. If you know a way to do this job please
> let me know.
>
> Thanks again
>
> Javier Martinez